

程序设计实习课程 QT 作业报告

一、程序功能介绍

1.注册、登录登出

用户注册时可以设置用户名、密码，后续登录时输入用户名密码即可登录。

2.创建任务

用户可输入任务内容、任务时间、任务类型（学习、生活、家庭、志愿等）、紧急程度、参与成员的姓名及邮箱等内容，并且自定义一个或多个提前一定时间的闹钟（详见功能三）。

3.自定义闹钟

用户在创建任务时可以选定提醒时间及形式，程序读取本地时间来发出提醒。支持对自己和其他成员的自动邮件提醒和对自己的本地弹窗和闹铃声音提醒。

4.任务查看

可视化界面列出所有任务，按颜色区分不同类型任务以及已完成、未完成和已过期任务。可以选择按截止时间或者紧急度排序。可以筛选出不同类型和状态的任务。

在此页面用户可更改某个任务的内容、截止时间以及提醒时间等设置，也可以改变任务状态（已完成或未完成）、删除某个任务。

5.总结与成就

用户可在此页面查看近期完成任务的总结（总计任务数量、完成任务数量、错过ddl 任务数量等），触发特定条件可以解锁一些成就（包括隐藏成就）。

6.设置

用户可更改自己的用户名、密码等信息，也可自己设置界面主体颜色、界面亮度等显示内容样式。

7. 日历模式

在日历模式中，用户可以十分方便地查看每日是否有任务（不同状态的任务依靠标记的颜色区分）以及任务的名称，同时可以一键跳转到任务查看页面直接定位到所选日期，查看任务的详细信息。

注：功能 1-6 与之前项目设计的功能相同，在实现原设计的基础上添加了功能 7，提高了项目的可用性。

二、项目各模块与类设计细节

前端页面设计：

对于上述功能中每个页面设计了一个类（`signup_window`、`login_window`、`settings_window`、`setnew_window`、`feedback_window`、`demo_window`、`calendar`），通过 `ui` 文件编辑页面样式，方便创建相应的页面。各页面之间较为独立，方便对某个页面进行调整。

`signup_window` 和 `login_window` 主要实现了几个按钮的槽函数，实现了登录和注册两个页面之间的切换（通过向 `login_and_signup` 类的一个成员发送响应信号），以及按住显示密码按钮可以展示已输入的密码信息。同时，在点击“登录”或“注册”按钮时对用户输入的用户名以及密码进行处理，并对错误情况进行弹窗提示。若成功注册则在提示后跳转到登录页面；若成功登录则自动关闭登录页面，打开主界面。

其他几个功能页面左侧按钮通过向 `mainwindow` 类的一个成员发送相应的信号实现各功能页面之间的切换。

`setnew_window` 中运用了 `QLabel`、`QLineEdit`、`QTextEdit`、`QComboBox`、`QTimeEdit`、`QDateEdit`、`QScrollArea`、`QPushButton`、`QGroupBox`、`QCheckBox`、`QSpinBox` 等多个 `ui` 类的成员，实现了任务各种内容元素的方便输入。在用户点击添加和提交按钮时将输入框中的内容传入后端函数实现新建任务，在用户有输入但未提交时点击切换页面、清空提醒邮箱、提醒时间或在用户提交任务时通过 `QMessageBox::question` 进行询问确定。通过正则表达式初步判定用户输入的

邮箱是否格式正确，进行相应提示。按照用户在提醒方式中的选择自动禁用或启用邮箱输入区以及提醒时间输入区，可以明显地提示用户所需输入的内容。对于已经添加的提醒邮箱和提醒时间分别用一个 `QScrollArea` 进行动态展示。每个提醒时间对应一个 `QFrame`，用 `QVBoxLayout` 进行布局。每个提醒邮箱对应一个 `QGroupBox`，包含邮箱用户的名字以及邮箱地址，用 `QHBoxLayout` 进行布局。在添加时将其加入对应的 `QScrollArea` 的下 `Widget` 之中。

`demo_window` 实现了对任务的展示，通过维护 `left_task` 和 `right_task` 两个变量维护左右展示的任务的 `id`，通过遍历任务数组进行两个变量的切换（点击下一页时向后寻找有效并且应该展示（即 `vaild` 和 `show` 变量均为 `true`）的任务，点击上一页时向前寻找），通过是否有 `right_task` 进行右侧展示页面的隐藏，并且通过 `next_task`、`pre_task` 判断现在是否为第一页或最后一页，对应禁用上一页或下一页按钮。并且这个实现可以在向下一页切换时直接将 `left_task` 变为 `next_task`，向上一页切换时将 `right_task` 变为 `pre_task`，将对四个 `id` 变量的维护统一起来。对于每个任务实现了完成任务、查看提醒/修改、删除任务按钮。通过点击完成任务可以改变任务状态到已完成，点击查看提醒/修改会弹出一个新的 `detailed` 对象窗口。点击删除任务时会调用对应的后端函数进行任务以及对应的提醒闹钟的删除，同时在删除后将原来任务的位置由后面的任务补位。此处涉及若为最后一页只有一个任务，删除后应自动调回前一页；删除任务可能导致下一页按钮由可用变为禁用等问题，实现中通过分类讨论正确维护了 `next_task`、`pre_task`、`left_task` 和 `right_task` 四个变量，保证了展示页面的正确性。在展示页面上部还实现了筛选器，用户可通过单选按钮实现排序方式的切换（排序中默认按截止时间顺序排序，并将已完成的任务放到最后），或通过多选按钮选择需要展示的任务类型和状态（通过调用 `filter` 函数将符合条件的任务的 `show` 成员变量改为 `true`，不符合条件的任务 `show` 成员变量改为 `false`）。

`feedback_window` 中将存储的相应的变量进行展示，利用了 `QLCDNumber` 类。通过相关函数判定成就是否完成，进行成就展示颜色的切换以及隐藏成就的展示。

`settings_window` 通过调用后端函数实现用户名和密码的更改，并且对用户名的重复等问题进行弹窗提示。通过 `QSlider` 进行页面亮度以及页面背景颜色的

调整。页面背景颜色利用一个 `label` 进行预览，点击确定后才会更改。页面亮度通过在整个页面上覆盖一个新的窗口实现，这个窗口会透传鼠标点击操作，背景颜色为黑色，其透明度会随着 `slider` 的拉动直接更改，实现视觉上亮度变化的效果。

`calendar` 页面使用了自带的 `QCalendarWidget` 并提升为 `CustomCalendar`。通过一个 `QScrollArea` 动态展示选定日期的任务名称和状态（通过 `on_calendarWidget_clicked` 槽函数进行刷新）。跳转到选定日期按钮通过发送带参数（选定日期）的信号跳转到 `demo_window` 并自动向后翻页到对应日期的第一个任务。这个文件中维护了一个 `map<QDate,int>` 类型变量，对所有任务的日期保存到 `key` 中，并利用二进制方法维护 `value`，描述了相应日期的任务类型。

创建的 `CustomCalendar` 的类是 `QCalendarWidget` 的派生类，重写了虚函数 `paintCell`，根据每个 `cell` 的日期在上面的 `map` 中查找并确定当日有的日期类型，在有任务的日期相应位置利用 `QPainter` 画 1~3 个对应颜色的圆点用于展示。

另外创建了 `detailed` 类，用于在任务查看页面查看页面详情，通过代码修改 `ui` 类成员的位置、大小实现了对于有不同提醒方式的任务展示的页面大小、内容位置不同，避免大片空白，照顾到页面的美观性。在此页面可以对任务名称、任务内容、截止时间、任务类型、紧急程度点击相应按钮调用后端相应函数进行修改，也可以添加或删除提醒邮箱以及提醒时间（这里的修改方式所用的 `ui` 类与新建任务中相似）。修改后会刷新展示页面。

而 `login_and_signup` 和 `main_window` 两个类主要实现了 `stackedWidget`，用于在登录和注册，以及程序几个功能页面之间进行切换。这两个类主要是接收其他具体页面发出的信号（例如 `goto_signup`、`login_success`、`goto_setnew`、`goto_feedback`、`go_exit` 等），并且连接到相应的槽函数，进行页面的切换，并根据情况调用相应页面的 `init` 函数进行页面的刷新，相较创建新窗口的切换方式视觉上更为流畅，并且可以保存上次离开页面时的信息。

前端所使用的背景图片以及按钮的背景等图片利用资源文件形式进行存储组织和调用。

后端函数实现:

后端函数主要在 `fun.h/fun.cpp`、`email.cpp` 文件中实现。后端中的任务和闹钟用数组存储实现。其中实现了 `user` 类、`task` 类和 `alarm` 类。`User` 类中主要成员变量为用户的用户名与密码以及用户的自定义颜色。`Alarm` 类中主要成员变量为其 `id`、对应的 `taskid`、提醒时间、提醒方式、存储位置。`Task` 类中主要成员变量为其 `id`、任务名称、任务内容、截止时间、紧急程度、提醒方式、存储对应所有 `alarm` 的 `vector`、提醒成员的 `vector`（容器成员类型为用结构体定义的 `member`，包括名字和邮箱）、任务类型、任务状态、存储位置以及 `show` 和 `valid` 两个变量。由于 `alarm` 和 `task` 用数组存储，如果删除时直接删掉后面的移动，那么会导致 `id` 与位置不对应，无法直接由 `id` 找到任务，因此对 `task` 和 `alarm` 分别设置了一个 `map`（`id_to_location`），用于将 `id` 转换为数组中的位置。并且为了减少对这个 `map` 的修改，设置了一个 `set`（`empty_location`），在删除任务或闹钟时，会将其 `valid` 成员变量改为 `false`，并且将其位置加入 `empty_location` 中，并且将 `id` 从 `map` 之中删去。添加新任务时首先查看 `empty_location` 是否为空，若为空则向后添加，否则从 `empty_location` 中选位置添加。

除了查看和修改私有成员变量的成员函数外，`alarm` 类还实现了 `do_alarm` 函数，用于根据提醒方式进行提醒，其中邮件提醒调用 `email.cpp` 中函数，弹窗利用 `QMessageBox`，声音提醒用 `QSoundEffect` 播放资源中的文件直到用户关闭弹窗。

提醒另外实现了一个优先队列，按照时间进行排序，每一分钟检查优先队列是否有提醒到时间，如果到时间则通过 `id` 找到相应的 `alarm`，调用 `do_alarm` 函数进行提醒，如果启动程序时有过时的提醒则直接删除。

`make_task` 和 `delete_task` 用于创建和删除任务，删除任务时遍历删除提醒。`delete_alarm` 用于删除提醒，从数组中删除方法与 `task` 类似，不用从优先队列中删除，因每个 `alarm` 的 `id` 唯一，到时间时会发现 `id` 没有对应的数组下标而直接丢弃。

`filter` 函数遍历 `tasks` 数组，因 `type` 使用二进制标志，可以用五位整数表示五个状态的所有叠加可能，利用位运算检查是否满足筛选条件。`update_task` 通过把所有的 `task` 截止时间与系统时间比对，自动更新过期任务状态，利用了

localtime 函数。sort_task 函数对 tasks 数组按时间或紧急程度排序，并更新 id_to_location 以及 empty_location。

save 和 load 函数用文件读写保存用户的所有数据。为此 user、task、alarm 类均重载了<<和>>运算符方便文件读写。

login 和 signup 函数进行登录和注册。注册时检查了用户名重复以及用户名为空或密码为空情况。登录时检查了用户名为空、密码为空、用户名不存在、密码错误情况。函数通过将对应的错误写入传进的 error 位置便于前端展示。

email.cpp 中实现了发邮件的功能，利用 WindowsSocket 进行网络通信。首先与 163 邮箱服务器（1.95.20.22:25）建立连接，然后根据 SMTP 协议将用户名和密码 base64 编码后发送（注：由于网易邮箱安全要求，密码为生成的客户端授权密码。为方便调试和验收，密码和邮箱硬编码在代码中，没有采用安全措施写在环境变量中，此授权密码将于 2025 年 7 月底过期），验证成功后根据格式发送信封和信件内容（收件人、主题、正文由参数传入，主题和内容中包括任务名称、任务内容以及完成状态），实现邮件发送功能。

计时器由多线程方法实现，在 main 函数中利用 QThread 建立了另外一个线程，主线程打开登录页面，而将一个 Worker 对象移至另一个线程运行 doWork 函数。Worker 类实现了一个 doWork 函数，其中的循环每隔一分钟向主线程发送一个 requestRing 信号，主线程接收到信号后会检查存有提醒的优先队列，判断是否需要发送邮件或者弹窗闹钟。

三、小组成员分工情况

王恩博：所有后端函数，登录注册页面的设计美化、槽函数实现，日历模式和 detailed 类的设计和槽函数实现

梁梓涵：其他所有前端页面的设计和美化，以及部分页面的槽函数实现

曾浩轩：其他所有前端页面的槽函数实现

共同进行了多次讨论和项目的 debug

四、项目总结与反思

本项目综合了前后端多项技术，包括多线程、邮件发送、类的编写组织、页面设计与美化等工作，通过小组合作完成本项目，了解了更多相关的知识，提升了较大型项目设计的工程经验。

在实现过程中仍存在 **github** 使用不熟练导致的版本管理问题，并且可以进一步在数据存储以及页面设计方面进行优化，例如用数据库存储用户数据、对页面进行进一步美化，加入按钮动态效果、扩展总结页面展示内容和成就种类等。